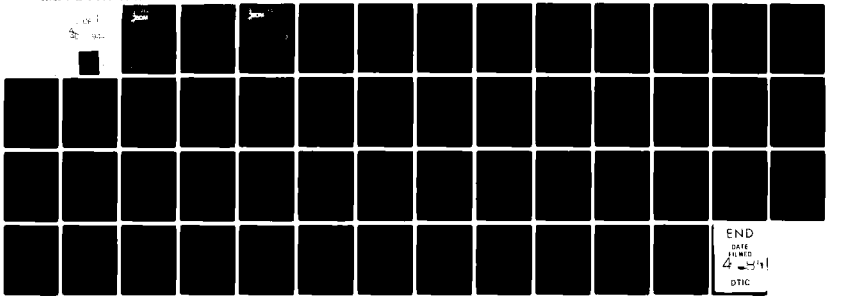
AD-A096 942 BDM CORP MONTEREY CA F/G 4/1
ENVIRONMENTAL MEASUREMENTS FOR MAGAT, BLM, AND STREX. INSTRUMEN--ETC(U)
FEB 81 C W FAIRALL, D E SPIEL N00014-78-C-0204
UNCLASSIFIED BDM/M-TR-0003-81 NL



END
DATE FILMED
4-81
DTIC

LEVEL II

THE BDM CORPORATION

AD A096942

DTIC
ELECTE
MAR 2 7 1981
S D
F

DTIC FILE COPY

81 3 06 067

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

LEVEL II (12)

THE BDM CORPORATION

P. O. Box 2019
2600 Garden Rd.
Monterey, California 93940
Phone (408) 649-3880

(11) 12 Feb [redacted] 81

(14) BDM/M-TR-0003-81

(12) 62

(9) Technical report

(15) N00124-78-C-0204, N00014-77-C-0

(6) ENVIRONMENTAL MEASUREMENTS FOR
MAGAT, BLM, AND STREX
INSTRUMENTATION AND DATA ACQUISITION PROGRAMS.

by

(10) C. W. FAIRALL AND D. E. SPIEL
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93940

AD A110162

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: BDM/M-TR-003-81 ✓ 2. GOVT ACCESSION NO. AD-A096 942	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): Environmental Measurements for MAGAT, BLM, AND STREX - Instrumentation and Data Acquisition Programs	5. TYPE OF REPORT & PERIOD COVERED: Technical 1981
	6. PERFORMING ORG. REPORT NUMBER: BDM/M-TR-003-81
7. AUTHOR(s): C. W. Fairall and D. E. Spiel	8. CONTRACT OR GRANT NUMBER(s): N00014-78-C-0204 ✓ and N00014-79-C-0088
9. PERFORMING ORGANIZATION NAME AND ADDRESS: The BDM Corporation, 2600 Garden Road, P.O. Box 2019, Monterey, CA 93940	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Work Order Nos. 422, 302 and 315
11. CONTROLLING OFFICE NAME AND ADDRESS: Dean of Research Code 012, Naval Post Graduate School, Monterey, CA 93940	12. REPORT DATE: February 12 1981
	13. NUMBER OF PAGES: 51
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office): Mr. Robin Simpson, ONR Representative, Stanford University, Room 165, Durond Aeronautics Building, Stanford, CA 94305	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
See Initial Distribution List Page 51

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
Page vii

DD FORM 1 JAN 73 1473 (Page 1) EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

FOREWORD

This report was prepared under Work Order Nos. 422, 302 and 315 of Contract Nos. N00014-78-C-0204 and N00014-79-C-0088, in support of the U.S. Naval Postgraduate School research project supported by the Naval Air Systems Command (Air 370) and the Naval Material Command (EO/MET). The work was performed in support of the Environmental Physics Group at NPS under the direction of Professors G. E. Schacher and K. L. Davidson.

TABLE OF CONTENTS

ABSTRACT

Instrumentation and calibration of certain systems used on aircraft and ships for atmospheric research at NPS are described. Computer programs for acquisition and analysis of aerosol and micrometeorological data are included.

A. INTRODUCTION

The Environmental Physics Group (EPG) at the Naval Postgraduate School (NPS) has participated in several field experiments in FY 80-81: the Marine Aerosol Generation and Transport Experiment (MAGAT), the Bureau of Land Management (BLM) tracer experiment and the Storm Transfer and Response Experiment (STREX). BDM personnel provided support to these efforts in the realm of instrumentation, calibration, data acquisition interfacing and programming, analysis and in-situ field participation. The purpose of this report is to document some of the instrumentation description, development and the data acquisition programming.

There are two experimental platforms (ship and aircraft), two types of data acquisition computers (HP9825-HPL and HP9835-BASIC) and two types of data (meteorological and aerosol). Because of various considerations, there are several combinations of these factors. For example, for MAGAT the ship used the HP9825 for meteorological data and the HP9835 for aerosol data while the aircraft used the reverse. For STREX a single HP9825 was used to obtain both types of data.

The instrumentation is described in Section B-1 and the programs are given in Section B-2.

B. INSTRUMENTATION

1. Shipboard Instrumentation

Most of the equipment used on board ship has been described previously (Houlihan, et al., NPS Report No. 61-78-001, 1978), so this part of the report will deal only with the ship roll-rate and heading sensors recently installed by BDM.

Ship Heading: On the RV/ACANIA a Librascope 7070 analog-to-digital converter has been connected to the gyroscope used for ship navigation. The converter is a mechanical device that senses the readout angular orientation. A buffer interfaces the serial output of the converter to a 16-bit interface to the computer.

Ship and Roll Rate: A mechanical pendulum geared to a potentiometer, placed on and pivoting about the roll axis of the ship, provides an analog signal proportional to the ship's roll angle. A simple RC network is used to differentiate this signal to yield an output related to roll rate. The rate of roll is useful, for example, in compensating for the effect of roll on the measurement of the standard deviation in wind direction. Errors introduced by vertical and lateral accelerations of the ship are minimized by placing the pendulum on the roll axis and midway along the length of the ship.

2. Aircraft Equipment

The aircraft is operated by Airborne Research Associates of Boston, Massachusetts. The turbulence equipment was installed by NPS, the remainder belongs to Airborne Research Associates. Most of the aircraft equipment is the same as on the ship. Besides the meteorological parameters, the aircraft also carries equipment for atmospheric electricity measurements, electric field and conductivity. This equipment will not be described here. For the equipment listed below most of the needed detail can be found in the shipboard system descriptions. The following will describe important differences and equipment that is unique to the aircraft. Equipment that is identical to shipboard will not be described.

Temperature and Wind Speed Fluctuations: Both utilize 4.5 μm tungsten wires rather than the platinum wire and films used on the ship. The wires are needed rather than the films because vortex shedding from the sensors distorted the signals at aircraft air speeds. The tungsten wires are much sturdier than the platinum wires which break frequently during aircraft operation.

Temperature fluctuations are measured with TSI Model 1044 DC bridges. When the turbulence sensors were first installed on the aircraft, excessive noise was encountered due to acoustic noise from the propeller. Three foot extensions were placed on each wing tip and the probes placed on the ends of the extensions. The increased distance from the prop (18 feet) solved the problem. The temperature fluctuation probes are separated by a vertical distance of .8 m for spatial filtering. The noise level of this system is about 3 mK rms.

Sea Surface Temperature: The radiometer is a PRT-5 as is on the ship, but its interpretation is slightly different. When the aircraft is low, (10 feet above the sea surface is one of the heights used) reflected radiation from the airframe is significant. This always raises the recorded temperature slightly and a correction must be applied.

Wind Speed and Direction: The relative air speed is determined by a MKS, Inc. capacitive differential pressure transducer, Model 223A. The aircraft normally flies with the average relative wind directly on the nose. The aircraft true ground speed and direction is determined with a Teledyne 711 Loran C navigation system. True wind speed and direction are calculated by the onboard computer. The air speed is accurate to 0.5 m/sec, the true wind speed accuracy depends on the averaging time but is about 1.0 m/sec for a two-minute run.

Air Temperature: Rosemount total temperature aircraft probe in a 102E housing. The probe is mounted in the window on the left side of the aircraft.

Dew Point: EG&G Model 137 aircraft system. The cooled mirror principle is used (similar to the ship system but specifically designed for aircraft use).

Altitude: Altitude is determined by sensing the pressure with a solid state device and also by radar altimeter. The pressure sensor is a National Semiconductor Type LX1702AN.

Microwave Refractive Index: Airborn microwave cavity refractometer supplied by the Naval Air Center (Indianapolis).

Aerosol Spectrum: Particle Measurement Systems, Inc. (PMS) type ASSAP aerosol spectrometer with PDS-200 controller. This unit was on loan from NOSC. Particles measured in four ranges of 15 bins each from 0.4 to 15 μm radius.

C. DATA ACQUISITION PROGRAMS

1. Ship

a. BLM-METEOROLOGY

```
0: "  ..  ....  1/3/81":
1: dev "t",701;dim V[30],V$[9],D[8],A$[27,4],P$[2,360]
2: dim L[3,9],F$[2,10],U,S[3],X[3,2],Z[3,2],K[3,5],B[2],T[3,10],P[10],D[5]
3: dim A[3,9],L[5,1],K[3,9],D$[3],I[3,9],Y[3],C[5,2],S$[10],U[11,3]
4: st: 14;trk 0;14k 1;7→X[1];20.5→X[2]
5: for I=1 to 3;for J=1 to 9;10(I-1)+J-1→K[I,J]→L[I,J];next J;next I
6: for J=7 to 9;40→K[3,J]→L[3,J];next J;40→K[2,3]→L[2,2]→K[1,7]→L[1,7]
7: 100.14→D[1];100.1→D[2];100.12→D[3];100.3→D[4];100.3→D[5]
8: "V1"→A$[1];"<V1>"→A$[2];"2T]"→A$[3];"C1"→A$[4];"Vic"→A$[5]
9: "L;"→A$[6];"T'"→A$[8];"3'"→A$[9];"V2"→A$[10]
10: "<V2>"→A$[11];"C2"→A$[13];"3/t"→A$[14];"3D"→A$[15]
11: "J"→A$[16];"P'D'"→A$[17];"<D>"→A$[18];"T01"→A$[19];"T02"→A$[20]
12: "T1"→A$[21];"T2"→A$[22];"Ts"→A$[23]
13: 2→X[3,1];10→X[3,5];for I=1 to 2;5→X[I,4];70→X[I,5];next I
14: "START":ent "MARK NEW PAPER?(yes)",D$;if D$="yes";jmp 2
15: ent "ENTER PAPER NO. TO BE MARKED",Y[3];cll 'mark';jmp 7
16: ser "LOAD DATA PAPER(cont)";stp
17: trk 0;LOF 0,Y[*];if Y[1]<50;jmp 4
18: trk 1;LIF 0,Z[*];if Z[1]<>0;jmp 3
19: ser "PAPER FULL,CHANGE PAPER.(cont)";stp
20: gto "START"
21: trk 0;lsn "PAPER",Y[3],"TRK",Y[2],", SKIP FILE",Y[1]+1;stp
22: ent "Enter calib?(y,n, or t=tone)",A$
23: if A$="t";cll 'recalData';jmp 1
24: if A$="n";jmp 3
25: if A$="y";jmp -3
26: fxd 1;cll 'CF';cll 'epsilon';cll 'temp';cll 'spin';cll 'Ctq'
27: fxd 2;"→D$;ent "Remove Sensor",D$;if D$="y";cll 'levsen'
28: "** READ IN DATA **":
29: "time":ent "Start time?(min.)",P;ent "Rec. interval?",U
30: fxd 0;lsn "Waiting for P=",P
31: ent 2%,cl0;red 709,P$[1];if flg1;cll 1;gto "time"
32: if val(P$[1,7,8])#2;jmp -1
33: "temp":P[1]→r1;ina K,I,C,E,F,A,S,2,U,3;0→X→F;1→J;r1→P[1]
```

```
34: fmt 2x,c10;red 706,T3[1]
35: val(P$[1,5,6])+val(P$[1,7,8])/60+val(P$[1,9,10])/3600→X
36: cll 'LORAN';P[7]→P[5];P[8]→P[6];cll 'DIR';cll 'AVEDEGS'(D,P[2],P[3])
37: "cycle":if P=30;cll 'short';ina 0;0→P
38: P+1→P;fmt c24;wrt 709,"VR3VA0VD5VF1V30V.0VT4VC0"
39: dsp "Print in",(T+J-val(T3[2,7,8]))mod60,"minutes,have=",X
40: for I=1 to 2;for M=1 to 3;if L[I,M]>39;gto "next"
41: fxd 0;"AI"&str(L[I,M])→V$
42: wrt 709,V$;fmt f;red 709,Z;Z+R[I,M]→R[I,M];1+R[I,1]→N[I,M]
43: if N=1 and M=6;Z+Q[4]→Q[4];jmp 3
44: if N=2 and M=4;Z+Q[5]→Q[5];jmp 2
45: if N=2 and M=6;Z+Q[7]→Q[7]
46: if M=2;Z+C[M+2,1]→C[M+2,1];1+C[M+2,2]→C[M+2,2];gto "next"
47: if M=4;Z+C[M,1]→C[M,1];1+C[M,2]→C[M,2]
48: if N#2 or M<4 or M>6;gto "next"
49: Z→T[I,M-3];1+C[11,M-3]→C[11,M-3];if M=6;Z+C[5,1]→C[5,1];1+C[5,2]→C[5,2]
50: "next":next M;next I;fmt c9;wrt 709,"VR1VA1V23"
51: fxd 0;for M=1 to 6;if L[3,M]>39;jmp 6
52: fmt f3.0;wrt 700,L[3,M]-10;wait 5;"AC"&str(L[3,M]-10)→V$;wrt 709,V$
53: fmt f;red 709,Z;1000Z→Z;Z+R[3,M]→R[3,M];1+N[3,M]→N[3,M]
54: if M=2;Z+Q[3]→Q[3];jmp 3
55: if M=4;Z+Q[2]→Q[2];jmp 2
56: if M=5;Z+Q[1]→Q[1]
57: next M;wrt 700,"C";1+X→X;1+J→J;if J<11;jmp 2
58: 1→J;1+P[4]→P[4];cll 'DIR';cll 'AVEDEGS'(D,P[2],P[3]);gto "forever"
59: if Xmod60#2;jmp 4
60: if X<50;jmp 3
61: cll 'LORAN'
62: cll 'JOGAY'(P[5],90-P[6],P[7],90-P[8])
63: "** FLAGS and TIMECHECK **":
64: if flg1;cfg 1;gto "time"
65: if flg2;cfg 2;cll 'levsen'
66: if flg3;cfg 3;cll 'temp'
67: if flg5;cfg 5;cll 'epsilon'
68: if flg6;cfg 6;cll 'main'
69: if flg9;cfg 9;gsb "SUBSUR SPARUS"
70: if flg10;jmp 3
```

```
71: if flag11;cfg 11;goto "mens"
72: call 'tirecheck'(T+U,J);if not flag12;goto "cycle"
73: cfg 12
74: disp "FLAG 11 FOR 2i";for i=1 to 20;beep;wait 1000;if flag11;jmp 2
75: next i;-1→A[1,8];goto "calculate"
76: ent "INPUT INVERSION HEIGHT IN METERS",A[1,8];cfg 11
77: goto "calculate"
78: "tirecheck":fmt 2x,c10;red 703,P$[2];val(P$[2,7,8])→r1
79: if r1>=p1mod60 and int(p1/60)=0;cfg 12;ret
80: if r1<p1mod60;ret
81: if int(p1/60)>0 and abs(r1-p1mod60)<=int(p2/2);cfg 12
82: ret
83: "** ACTIVATE OR DEACTIVATE **":
84: "levsen":ent "Activate(a),Deactivate(d)or(f)",Q$;if Q$="f";ret
85: if Q$#"a" and Q$#"d";beep;jmp -1
86: ent "Sensor Name(see list)",S$;if S$="f";jmp -2
87: for N=1 to 3;for I=1 to 9;if S$=A$[D(N-1)+I];jmp 2
88: next I;next N;beep;jmp -2
89: if Q$="d";41→L[N,I];jmp -3
90: X[I,3]→L[I,F];jmp -4
91: "SENSOR STATUS":prt "SENSOR STATUS";prt T$[2];prt "DEACTIVATED"
92: for N=1 to 3;for I=1 to 9;if L[I,3]>40;prt A$[D(N-1)+I]
93: next I;next N;ret
94: "** CALIBRATION **":
95: "ersilon":for J=1 to 2;X[J,1]→r1;0→r2;dsp "LEVEL",J,"Vo^2",r1
96: ent "",X[J,1];if X[J,1]#r1;2→r2
97: X[J,2]→r1;dsp "LEVEL",J,"3",r1;ent "",X[J,2];if X[J,2]#r1;2→r2
98: X[J,3]→r1;dsp "LEVEL",J,"GAIN(epsln)",r1;ent "",X[J,3];if X[J,3]#r1;2→r2
99: if r2>1;0→N[J,1]→R[J,1]
100: next J;ret
101: "ship":P[1]→r1;dsp "SHIP'S SPEED(kts)",r1;ent "",P[1];if P[1]=r1;ret
102: 0→P[2,6]→N[2,6];for P=1 to 2
103: 0→N[P,1]→R[P,1]→N[P,4]→R[P,4];next P;ret
104: "temp":dsp "Pt TEMP LEVEL 1",D[1];ent "",D[1];dsp "Pt TEMP LEVEL 2",D[2]
105: ent "",D[2];dsp "Pt TEMP SURFACE",D[3];ent "",D[3];ret
106: "ReCalDate":trk Y[2];lif 0,Y[*]
107: lbf Y[1],P$,L[*],T$,0,S[*],R[*],N[*],X[*],M[*],T[*],P[*],D[*],H[*],E[*]
```

```
108: ret
109: "** CALC IBT PARAMETERS **":
110: "calculate":for I=1 to 3;for J=1 to 3;if L[I,J]>39;0→R[I,J];jmp 2
111: P[I,J]/X[I,J]→R[I,J]
112: next J;next I
113: 100R[2,6]→3[1,7];3[1,7]→3[1,7]→P
114: if R[2,4]=0;10R[1,4]→V;jmp 2
115: 10R[2,4]→V
116: deg;√(V↑2+P[1]↑2-2V*P[1]*cos(R))→r1;.514r1→3[3,7];Vsin(P)/r1→S
117: (-P[1]+Vcos(R))/r1→O;if O>0;asn(S)→3[2,7];jmp 3
118: if S>0;acs(O)→3[2,7];jmp 2
119: 180-asn(S)→3[2,7]
120: (3[2,7]+P[2])→3360→3[2,7]
121: .00395→r1;(P[3,5]-D[3])/r1D[3]→3[1,6]
122: for I=3 to 4;if L[3,I]>39;jmp 2
123: (R[3,I]-R[I-2])/r1D[I-2]→3[I-2,2]
124: next I;.00399→r1;for I=1 to 2;if L[3,I]>39;jmp 2
125: (R[3,I]-R[I+3])/r1D[I+3]→3[I,1]
126: next I;10(R[1,6]-1)→3[3,8];if R[1,5]<.01;jmp 2
127: 396.2/R[1,5]→3[2,8]
128: cll 'vapcal';if L[3,5]>39;jmp 2
129: 273.15+3[1,6]→Y;cll 'Q';Q→3[3,3]
130: for I=1 to 2;if L[3,I]>39;jmp 3
131: 273.15+3[I,1]→Y;cll 'Q';Q→3[I,3]
132: 273.15+3[N,2]→Y;cll 'Q';100R[N,3]/Q→3[N,4]
133: next N;jmp 2
134: "Q":.625*10↑(23.84-2948/Y-5.03log(Y))→Q;ret
135: if L[1,3]<40;(20.7R[1,3]/W[1,1]W[1,2])↑2→3[2,6]
136: for N=1 to 2;if L[N,1]>39;jmp 3
137: 618(5.14R[N,4])↑.5*(R[N,1]R[N,2]/X[N,2]X[N,3])↑3→Q
138: Q/(X[N,4]↑(-.667)-X[N,5]↑(-.667))↑1.5→4[I,5]
139: next N;if L[1,9]>39 or L[1,8]>39;jmp 5
140: 9.08(5.14P[2,4])↑(-.667)/(X[3,4]↑(-.667)-X[3,5]↑(-.667))→V
141: P[2,9]*X[3,3]/4.35→Z
142: .0036*20*W[3,1]W[3,2]→Y;if 3[2,3]≠0;.78*R[2,9]X[3,3]*X[3,2]/3[2,3]→Z
143: V(1[1,9]/Z)↑2→3[1,9];V(P[1,8]/Y)↑2→3[2,3];10V*R[2,3]/(Y*Z)→3[3,9]
144: 9.8*10↑(-7)(3[2,6]+.113.3[3,9]+.00324[1,9])↑.5→3[3,6]
```

```
145: "** PRINT DOP **":
146: "print":fmt c16,5x,z;wrt "P","Date :"&T$[2,1,2]&"/"&T$[2,3,4]&"/80"
147: fmt c6,f3.0,4x,c7,f3.0,z;wrt "P","Tape#",Y[3],"Track#",Y[2]
148: fmt 4x,c6,f4.0;wrt "P","File#",Y[1]+1
149: fmt 1,c17,4x,c16,f4.0,z
150: wrt "P.1","End time: "&T$[2,5,6]&":"&T$[2,7,8],"Averaging time:",J
151: fmt c8,1x,c17,f4.0,10x,c5;wrt "P","minutes","Number averages:",K,"BLT"
152: fmt c17,f10.4,c15,f10.4,c14,2/
153: wrt "P","Initial position:",P[5],"Deg Longitude",P[6],"Deg Latitude"
154: fmt 87x,c18;wrt "P","Channel Assignment"
155: fmt 11x,c5,16x,c9,11x,c14,z;wrt "P","Ship","True Wind","Relative Wind"
156: wtb "P",13;cll 'space'(7)
157: for I=1 to 3;for J=1 to 13;wtb "P",95;next J;cll 'space'(10);next I
158: fmt /,74x,6f5.0,z;wrt "P",L[1,1],L[1,2],L[1,3],L[1,4],L[1,5],L[1,6]
159: fmt 3f5.0;wrt "P",L[1,7],L[1,8],L[1,9]
160: fmt 4x,c11,f5.1,7x,c11,f5.1,7x,c11,f5.1,z;M[3,7]/.514→P
161: wrt "P","Speed(kts):",P[1],"Speed(kts):",P,"Speed(kts):",10R[2,4]
162: fmt 4x,6f5.0,z;wrt "P",L[2,1],L[2,2],L[2,3],L[2,4],L[2,5],L[2,6]
163: fmt 3f5.0;wrt "P",L[2,7],L[2,8],L[2,9]
164: fmt 7x,c8,f5.0,10x,c8,f5.0,10x,c8,f5.0,z
165: wrt "P","Heading:",P[2],"Heading:",M[2,7],"Heading:",M[1,7]
166: fmt 3x,6f5.0,z;wrt "P",L[3,1],L[3,2],L[3,3],L[3,4],L[3,5],L[3,6]
167: fmt 3f5.0,2/;wrt "P",L[3,7],L[3,8],L[3,9]
168: fmt 40x,c12,/;wrt "P","RAW VOLTAGES"
169: fmt 1,4x,c5,4x,c4,5x,c8,3x,c7,2x,c10,3x,c8,z
170: wrt "P.1","Level","Cups","Velocity","Delta T","Hot film","Humidity"
171: fmt 2x,c11,4x,c6;wrt "P","Temperature","SigmaV"
172: fmt 6x,c,5x,c6,6x,c3,8x,c3,7x,c6,6x,c4,7x,c4,z
173: wrt "P","#","ave. U","RMS","RMS","ave.V","Ohms","Ohms"
174: fmt 25x,c7,13x,c5,z;wrt "P","Miscel.","Lynan"
175: wtb "P",13;for I=1 to 88;wtb "P",95;next I;fmt /;wrt "P"
176: fmt 63x,c3,f8.3,3x,c2,f7.3,6x,c6,f8.0,6x,c5,f8.3
177: wrt "P","Ts:",R[3,5],"WD",E[5,1]/C[5,3],"Zi(m):",M[1,8],"Orms:",R[1,9]
178: fmt 3,3x,c2,f7.3,7x,c5,f8.2,6x,c5,f8.3
179: fmt 1,2x,f5.0,f10.2,f11.3,z;for I=1 to 2
180: wrt "P.1",I,R[I,4],R[I,1]
181: if L[1,3]<41;cll 'DP';jmp 2
```

```
132: cll 'space'(12)
133: fmt f11.2,z;wrt "P",R[1,2];if L[3,M]<4;cll 'P';jmp 2
134: cll 'space'(11)
135: fmt f12.3,z;wrt "P",R[3,M+2]
136: if I=1;wrt "P.3"," '",E[5,2]/E[5,3],"  UU:",R[2,6],"Dave:",R[2,9]
137: if M=2;wrt "P.3","U2",100E[2,1]/E[2,3]," Vis:",R[1,5],"Trms:",R[1,8]
138: next M;fmt 77x,c2,f7.3,7x,c5,f8.2,6x,c5,f8.3
139: wrt "P"," '",E[2,2]/E[2,3],"  IR:",R[1,6],"  TQ:",P[2,8]
140: if L[1,9]<40;if L[1,3]<40;100R[2,3]/R[1,9]R[1,3]→r1
141: fmt 78x,c1,f7.3,27x,c4,f8.3,2/;wrt "P"," ",E[5,3],"Rtq:",r1;jmp 3
142: "DP":fmt f11.3,1x,z;wrt "P",R[M,3];ret
143: "P":fmt f11.2,z;wrt "P",R[3,M];ret
144: fmt 40x,c13,/;wrt "P","MICROJET DATA"
145: fmt 4x,c5,9x,c1,8x,c2,9x,c1,9x,c1,8x,c2,8x,c1,6x,c7,9x,c4,7x,c6
146: wrt "P","Level","Z","PD","T","a","RE","Q","epsilon","CP*2"," Sigma"
147: fmt 6x,c1,10x,c3,6x,c3,6x,c3,5x,c6,6x,c3,4x,c7,2x,c11,3x,z
148: wrt "P","#","(m)","(C)","(C)","(q/Kq)","(*)","(m/sec)","(m*2/sec*3)"
149: fmt c11,z;wrt "P","(c*2/r*2/3)";fmt 14x,c8,z;wrt "P","Stq Data"
150: wtb "P",13;for I=1 to 108;wtb "P",95;next I;fmt /;wrt "P"
151: fmt c13,f10.1,11x,f10.2,z;wrt "P","Surface",0,1[1,6]
152: if 1[1,6]=0;0→1[3,3];cll 'space'(11);fmt f9.1,z;wrt "P",99.9;jmp 2
153: fmt f10.2,1x,f9.1,z;wrt "P",M[3,3],99.9
154: fmt 39x,c2,f7.2,6x,c5,e10.2;wrt "P","dD",S[1],"Cm*2:",1[1,9]
155: fmt 1,f7.0,3x,f10.1,1x,z;fmt 2,4f10.2,f8.1,e14.2,z
156: fmt 3,15x,f10.2,20x,f8.1,e14.2,z;fmt 4,e14.2,3x,c2,f7.2,6x,c5,e10.2
157: fmt 5,15x,c3,f7.2,6x,c5,e10.2
158: for I=1 to 2;wrt "P.1",S,S[I]
159: if L[3,M]>39;wrt "P.3",M[1,2],5.1MP[1,4],S[1,5];jmp 2
160: wrt "P.2",M[M,1],M[M,2],M[1,3],M[1,4],5.1MP[1,4],S[1,5]
161: if I=1;wrt "P.4",M[2,5],"U1",S[2],"Ct*2:",M[2,9];jmp 3
162: if I=2;wrt "P.5","U2",S[3]," Ctq:",M[3,9];jmp 2
163: fmt 115x,c3,e10.2;wrt "P","Cn:",M[3,5]
164: next M
165: "** CALC & PRT SCALE **":
166: fmt 3/,40x,c13,/;wrt "P","SCALING PARAMETERS"
167: fmt 5x,c6,5x,c3,6x,c2,3x,c2,3x,c2,3x,c2,7x,c3,7x,c3,7x,c2,9x,c2,11x,c1
168: wrt "P","Method","L/L","Pi","u*","T*","q*","-wr","-wq","w*","h*","t"
```

```
219: fmt 37x,c7,3x,c7,3x,c6,22x,c7,18x,c5,x
220: wrt "1","(m/sec)","(deg.)","(m/sec)","(m/sec)","(min)"
221: wtr "r",13;for i=1 to 115;wtb "r",95;next i;fmt /;wrt "r"
222: fmt 1,c10,2e11.3,5f10.3,3;fmt t2,f10.2,14,f10.3,3x,f8.0
223: "COLA":.33[3,7]→y;iL L[3,1]>39;i[2,3]→y;jmp 2
224: if L[3,2]>39;.[1,3]→y;jmp 2
225: (.[2,3]+.[1,3])/2→y
226: if L[3,2]>39;.[1,2]→y;jmp 3
227: if L[3,2]>39;.[1,2]→y;jmp 2
228: (.[1,2]+.[2,2])/2→y
229: if V<2.2;.001*1.03*y*(-.15)→r1;jmp 4
230: if V<5;.001*(.77+.235*y)→r1;jmp 3
231: if V<8;.001*(.37+.317*y)→r1;jmp 2
232: .0011*(1.24+.235*y)→r1
233: .35/r1*.5→r2;.35*1.35/.0013*.5/r1→.
234: .35/238*.0013*.5/r1→.
235: 1e-4*(e^(-1.1+.89*1*(0+273.15)*(0-.[3,3]))/V^2→z
236: r=.sqrt(.pi(.)*z/(1-r/r2)*2)→r;for .=1 to 3;disp .,r,z
237: (1-.^2)*2/(1-r/r2)*2→r;for .=1 to 3;disp .,r,z
238: if z<(1-r)>.001*sin(.)→.;jmp -5
239: .→r[1,1];cll "plot",(.)*.35/.*v/(1-.^2)→r[1,3]
240: cll "plot",(.)*.35*1.35/r2*(0-.[1,3]+.1)/(1-r/r2)→r[1,4]
241: .35*1.35/r2*(0-.[3,3])/(1-r/r2)→r[1,5];cll "plot"(5);r→r[1,2]
242: cll "scaling"(1)
243: wrt "plot","scale",.[1,1],.[1,2],.[1,3],r[1,4],r[1,5],r[1,6],r[1,7]
244: for 2,5,1,e14.2,f10.1;wrt "",.[1,3],r[1,9],r[1,10]
245: "FUNCTION":10/.→y;for .=1 to 2
246: if .[1,5]#0;cll "plot"(z[1,1]/2);(.3*z[1,1]/2)^.333+.→y;jmp 2
247: next .;fmt 2(2,6)#0;cll "plot"(z[1]/2);(1[2,5]z[1]^.567/2)^.5+y→y;x+1→x
248: if .=0;0→r[3,3];jmp 2
249: r/.→r[3,3]
250: if .=3;.→r[3,4];jmp 2
251: (x/3)sgn(r[1,4])*1[3,4]
252: if 1[1,2]=3;.→r[3,5];jmp 2
253: cll "plot"(z[1]/2);sgn(r[1,5])*(1[1,2]*z[1]^.567/2)^.5/1.3→r[3,5]
254: if r[3,3]=3;.→r[3,1]→r[3,2];jmp 2
255: .119(r[3,1]+.21r[1,5])/r[3,3]^2→r[3,1];cll "plot"(r[3,1]);r→r[3,2]
```

```
256: cll 'scaling'(3)
257: wrt "P.1","Para",P[3,1],P[3,2],P[3,3],P[3,4],P[3,5],P[3,6],P[3,7]
258: wrt "P.2",P[3,8],P[3,9],P[3,10]
259: "** MISC PRT**":
260: fmt 3/,40x,c22,/;wrt "P","CALIBRATION PARAMETERS"
261: fmt 23x,c7,33x,c2,13x,c11;wrt "P","Epsilon","CT","Temperature"
262: fmt 15x,c4,7x,c1,3x,c1,7x,c2,6x,c2,11x,c1,8x,c1,14x,c2,12x,c7,z
263: wrt "P","Mo*2","Ts","S","F1","Fu","R","S","Do","Diesel."
264: wtb "P",13;for i=1 to 92;wtb "P",95;next i;fmt /;wrt "P"
265: fmt 10x,c5,f3.1,f9.0,2f8.0,5x,f8.1,f9.0,2x,c7,f8.4,z
266: wrt "P","Lysan",X[3,2],X[3,3],X[3,4],X[3,5],W[3,1],W[3,2]," Surf. ",D[3]
267: fmt 5,3x,c4,f7.0,1x,c3;wrt "P.5"," Si:",N[1,3],"(m)"
268: wtb "P",10;fmt 2x,f3.0,f3.1,4x,f6.1,f9.0,2f3.0,5x,f3.1,f9.0,7x,f10.4,z
269: for i=1 to 2
270: wrt "P",1,X[1,1],X[1,2],X[1,3],X[1,4],X[1,5],W[1,1],W[1,2],D[M]
271: if i=1;wrt "P.5","Vis:",N[2,3],"(m)"
272: if i=2;fmt 6x,c4,f9.2,1x,c3;wrt "P"," IR:",I[3,3],"(C)"
273: next i;0→r2;P[4]→r3;for i=1 to r3;itf(F$[2,21-1,21])→r1
274: r1-r2→r4;abs(r4)→r5;if r5>180;(r4/r5)(r5-360)→r6;(r6/3+r2)mod360→r2;jmp 2
275: r2+r4/i→r2
276: next i;0→r4;for i=1 to r3;abs(itf(F$[2,21-1,21])-r2)→r1
277: if r1>180;r1-180→r1
278: r4+r1↑2→r4;next i;√(r4/(r3-1))→r1
279: fmt 2/,c13,f7.2,c15,f4.1,c16,f6.1,z
280: wrt "P","Cor sig(WD) =",r1,"Loran speed =",P[9],"Loran course =",P[10]
281: fmt 10x,c13,f8.3;wrt "P","Standard R =",R[3,6]
282: if flg3;fmt /,c11;wrt "P","HIT ON TAPE";gto "increment"
283: "** TAPE **":
284: "tape":trk Y[2];1df 0,Y[*];Y[1]+1→Y[1];rcf 0,Y[*];Y[1]→M
285: rcf 1,F$,L[*],TS,U,S[*],R[*],W[*],X[*],Z[*],T[*],P[*],D[*],V[*],E[*]
286: if Y[1]<50;rew;jmp 3
287: if Y[2]=0;rew;1→Y[2];trk 1;0→Y[1];jmp 2
288: dsp "TAPE",Y[3],"IS FULL.";for i=1 to 20;beep;wait 500;next i;gto "START"
289: "increment":wtb "P",12
290: if flg10;cfg 10;fxd 0;cll 'timecheck'(T+U-5,J);if not flg12;gto "meas"
291: (T+U)mod60→T;fxd 0;gto "meas"
292: "** OUTPUT SUBS **":
```

```
293: "space":if p1<0;jmp 2
294: wtr "r",32;jmp 2((p1-1→p1)=0)
295: wtr "r",8;jmp 2(p1+1→p1)=0
296: ret
297: "Lic U":if p1<0;(1-15p1)↑(-.25)→U;ret
298: 1+.7*p1→U;ret
299: "Pl":if p1<0;4.9(1-7p1)↑(-.667)→P;ret
300: 4.9(1+2.4p1↑.667)→P;ret
301: "L U":if p1<0;(1+.5abs(p1)↑.667)↑1.5→P;ret
302: (1+2.5p1↑.8)↑1.5→P;ret
303: "Spit":if p1>0;-4.7*p1→U;ret
304: (1-15p1)↑.25→p2
305: 2*ln((1+p2)/2)+ln((1+p2↑2)/2)-2*atn(p2)+1.04→U;ret
306: "Spit":if p1>0;-6.5*p1→P;ret
307: (1-9*p1)↑.5→p2
308: 2*ln((1+p2)/2)→P;ret
309: "Rich":if p1<0;.74p1(1-15p1)↑.5/(1-13p1)↑.33→P;ret
310: p1(.74+4.7p1)/(1+4.7p1)↑2→P;ret
311: "profile":p1+R→R;p2+Q→Q;X+1→X;p1*p2+W→W;p1↑2+Y→Y;ret
312: "regression":.35(P*Q-X*W)/(R↑2-X*Y)→P;ret
313: "mark":0→Y[1];for I=0 to 1;rew;trk 1;mrk 1,50;I→Y[2];rct 0,Y[*]
314: trk 61,2200;next I;trk 0;ret
315: "scaling":-T[p1,3]T[p1,4]→T[p1,6];-T[p1,3]T[p1,5]→T[p1,7]
316: (.034abs(T[p1,6])M[1,8])↑(1/3)*sgn(p1)→T[p1,8]
317: if T[p1,8]=0;0→T[p1,9];0→T[p1,10];ret
318: T[p1,6]/T[p1,8]→T[p1,9];abs(M[1,8]/60T[p1,8])→T[p1,10];ret
319: "****        HOT FILM CAL SUBROUTINES        ****":
320: "avevar":for I=1 to 5;if C[I,2]=0;jmp 2
321: C[I,1]/C[I,2]→r1;r1+E[I,1]→E[I,1];r1↑2+E[I,2]→E[I,2];1+E[I,3]→E[I,3]
322: next I;if U[11,3]=0;ina U,C;ret
323: deg;for M=2 to 3;0→r0;for I=1 to U[11,3];10U[I,1]→r1;10-U[I,3]mod360→r2
324: 61.03U[I,2]→r3;if I=2;0→r3
325: r1sin(r2)-r3→r4;r1cos(r2)-P[1]→r6
326: √(r1↑2+r3↑2+P[1]↑2-2r1(r3sin(r2)+P[1]cos(r2)))→r5
327: if r4>=0;acs(r6/r5)→r7;jmp 3
328: if r6>=0;360+asn(r4/r5)→r7;jmp 2
329: 180-asn(r4/r5)→r7
```

```
330: r7-r0→r1;abs(r1)→r2
331: if r2>=180;(sgn(r1)(r2-360)/I+r0)mod360→r0;jmp 2
332: r1/I+r0→r0
333: next I;fti ((r0+.5)mod360)→P$[M-1,2P[4]-1,2P[4]];next "";ina C,U;ret
334: "varcal":for I=1 to 2;if L[I,4]>33;jmp 3
335: if E[I,3]<2;0→E[I,1];jmp 2
336: 5.14*((E[I,2]-E[I,1]↑2/E[I,3])/(E[I,3]-1))↑.5→E[I,1]
337: if L[I,2]>33;jmp 4
338: if E[I+2,3]<2;jmp 3
339: 2+I→K;((E[K,2]-E[K,1]↑2/E[K,3])/(E[K,3]-1))↑.5→E[K,1]
340: if E[I,1]≠0;if E[K,1]≠0;4*E[K,1]/E[I,1]*X[I,2]*√(5.14*P[I,4])→X[I,2]
341: next I;if E[5,3]>1;108*((E[5,2]-E[5,1]↑2/E[5,3])/(E[5,3]-1))↑.5→S[1]
342: E[1,1]→S[2];E[2,1]→S[3]
343: ret
344: "AVGDEG":p1-p2→p1;abs(p4)→p5
345: 1+p3→p3;if p5>=180;(p4/p5)(p5-360)→p5;(p5/p3+p2)mod360→p2;jmp 2
346: p2+p4/p3→p2
347: ret
348: "COURSCHG":dsp "COURSE CHANGE";for I=1 to 10;beep;wait 1000;beep;next I
349: dsp "Course change",P[2],"to",C,".Cont";if Flg0;jmp 4
350: ctr
351: ent "'c'to continue  'b' begins over",Q$;if Q$="c";ret
352: if Q$≠"b";beep;jmp -1
353: fmt 2x,cl0;red 708,P$[1];(val(P$[1,7,8])+1)mod60→T;gto "meas"
354: "AVGAV":val(T$[2,5,6])+val(T$[2,7,8])/60+val(P$[2,9,10])/3600→p5
355: (p5-A)mod24→p5
356: beep;acs(cos(p4)cos(p2)+sin(p4)sin(p2)cos(p3-p1))→p5
357: (3432.45+.129p4)p6*2*π/360,p5→P[9]
358: acs((cos(p4)-cos(p2)cos(p6))/sin(p2)sin(p5))→p7;p7mod360→p7
359: if p7<=90 and p3<p1;p7→P[10];jmp 4
360: if p7<=90;360-p7→P[10];jmp 3
361: if p3<=p1;p7→P[10];jmp 2
362: 90+p7→P[10]
363: P[10]mod360→P[10];ret
364: "DIR":wtb 4,2;wait 1;band(shf(cmprdb(4),2),32767)→p2;wtb 4,0
365: 0→p3→p1;for N=0 to 6;bit(p3+N,p2)→p4;p1+2↑N*p4→p1
366: if p4=1;0→p3;jmp 2
```

```
357: 0→n3
358: next i;127-n1→n1;(17.5+n1)*3.12/128*360→D;ret
359: "LORAN":if bit(0,rdb(4))=0;ret
370: wtb 4,192+32;wtb 4,192;if bit(1,rdb(4))=1;jmp 0
371: for i=1 to 80;rdb(4)→V[i];next i
372: if bit(1,rdb(4))=1;jmp -2
373: 0→n1→n4→n5;for i=0 to 7;if V#4;1+n1→n1
371: 10*(n1-5)→n3;n4+'bcd'(7+4*i)*n3→n4;n5+'bcd'(47+4*i)*n3→n5;next i
375: n4(1-2V[31)→P[8];n5(1-2V[17])→P[7];wtb 4,0
376: for i=7 to 8;abs(P[i])→n4;int(n4)→n5;100frc(n4)/60→n6
377: n5+n6→P[i];next i;ret
378: "bcd":V[n1]+2V[n1+1]+4V[n1+2]+8V[n1+3]→n2;if n2=15;0→n2
379: ret n2
330: "short":for i=1 to 8;D[i]/F→D[i];next i
381: 1080[7]mod360→R→D[7];100[5]→R→D[5]
382: deg√(R^2+P[1]^2-23*P[1]*cos(R))→r1;r1→D[6];asin(R)/r1→R
383: (-P[1]+Rcos(R))/r1→D;if D>0;acs(D)→D[8];jmp 3
384: if R>0;acs(D)→D[8];jmp 2
335: 180-asn(D)→D[8]
336: (D[8]+P[2])mod360→D[8]
387: (D[1]-D[3])/.003855[3]→D[1];10(D[4]-1)→D[4]
388: (D[2]-D[2])/.003855[2]→D[2];(D[3]-D[5])/.003891D[5]→D[3]
389: fxd 4;fmt 5x,c9,f11.4,5x,c9,f11.4,/
390: wrt 5,"latitude",P[8],"longitude",P[7]
391: fxd 0;fmt 5x,c13,f6.0,z;wrt 5,"ships heading",P[2]
392: fxd 1;fmt 6x,c11,f6.1,2/;wrt 5,"ships speed",P[1]
393: fxd 2;fmt 7,10x,c15,f10.2,1x,c3,2/
394: wrt 5.7,"sea temperature",D[1],"(C)";wrt 5.7,"air temperature",D[2],"(C)"
395: wrt 5.7,"dew temperature",D[3],"(C)";wrt 5.7," ir temperature",D[4],"(C)"
396: fxd 1;wtb 5,10;fmt 9,10x,c17,f10.1,2x,c5,f10.1,2x,c6,/
397: wrt 5.9,"wind speed (knts)",D[5],"(rel)",D[5],"(true)"
398: wrt 5.9,"wind direction  ",D[7],"(rel)",D[8],"(true)"
399: ret
*4165
```

b. MET-AEROSOL STREX

```
 0: " READS SPEC":sfq 11
 1: dim A$[15,41],B$[128],I3[60],P$[480],U$[60],Y$[24],C,E,J,Z[11]
 2: dim C$[1330],G$[10],G$[3]
 3: dim A[10],B[10],C[3],D[3],F[8,15],F[10,10],G[10],H[4,8],K[3,5]
 4: dim L[3,5],M[8],O[8,15],P[3,5],R[8,16],S[8,15],T[8,15]
 5: dim U[3,5],V[2,3],W[2,15],X[6],Z[6]
 6: 35→B[1];111→B[2];43→B[3];0→B[4];64→B[5];42→B[6];37→B[7];38→B[8]
 7: for J=4 to 5;40→K[3,J]→L[3,J];next J
 8: for J=1 to 5;1-1→K[1,J]→L[1,J];next J;5→K[2,1]→L[2,1];6→K[2,2]→L[2,2]
 9: 10→K[2,3]→L[2,3];11→K[2,4]→L[2,4];15→K[2,5]→L[2,5]
10: for J=1 to 3;6+1→K[3,J]→L[3,J];next J
11: 39.5939→D[1];99.4823→D[2];39.4433→D[3]
12: "ssp1"→A$[1];"Air"→A$[2];"Stir"→A$[3];"Ssp3"→A$[4];"<V>"→A$[5]
13: "Irr"→A$[6];"P"→A$[7];"IR1"→A$[8];"IR2"→A$[9];"Vrms"→A$[10]
14: "Dev"→A$[11];"Fair"→A$[12];"Bad"→A$[13]
15: .028→W[1,1];.217→W[1,2];.632→W[1,3];1.177→W[1,4];1.734→W[1,5]
16: 2.345→W[1,6];3.05→W[1,7];3.787→W[1,8];4.5→W[1,9];5.393→W[1,10]
17: 6.133→W[1,11];6.903→W[1,12];7.552→W[1,13];8.36→W[1,11];9.012→W[1,15]
18: 9.15→W[2,1];8.54→W[2,2];7.93→W[2,3];7.32→W[2,4];6.71→W[2,5]
19: 6.1→W[2,6];5.49→W[2,7];4.88→W[2,8];4.27→W[2,9];3.66→W[2,10]
20: 3.05→W[2,11];2.44→W[2,12];1.83→W[2,13];1.22→W[2,14];.61→W[2,15];28→Z[1]
21: wtb 9,0;rew 709;rew 711;rew 704;110 7;715→J;rev "1",1;lak 1;buf "0",A$,3
22: rew 722;wrt 704,"A55";wrt 722,"E1P4P3A0"
23: "Formats":fmt 0,10x,z
24: fmt 1,f3.0
25: fmt 2,cl,f3.0,z
26: fmt 3,"Date",f3.0,"/",f2.0,"/",f2.0,"  Time ",f4.0,":",f2.0," (PST)."
27: fmt 4,"Averaging time = ",f4.1," minutes","    No. averages ",f4.0
28: fmt 5,"Probe voltage  A = ",f6.3," volts"
29: fmt 6,e10.2,z
30: fmt 7,"Tape #",f3.0,"  File #",f3.0,"          SPEX"
31: fmt 8,"Polynomial of order ",f2.0,z
32: fmt 9,215.7,z
33: "RESET":dsp "SET ALL FLAGS (CONT).";stp
```

```
31: if flg10;clg 10;jsb "tapemrk"
35: if flg4;clg 4;asb "SPACES"
36: 7→C;if flg5;clg 5;ent "NUMBER OF RECORDS?",C
37: C+1→I;redim A[I],B[I],F[I,I]
38: 30→T;int(BT)→B;if flg5;clg 5;ent "AVERAGE TIME?",T;int(BT)→Z
39: dsp "ENTER DATA CARD (CONT).";stp
40: REM D,J,C[*]
41: dsp "PUNCH OR PRINTER(CONT).";stp
42: wtb 2,27,69,27,84,32,32,32,32,27,77,27,76,15,0,14
43: wtb 2,27,70,4,48,6,26
44: 3→X[1]→X[2]→X[3];1→X[5]→X[6];15→X[4];0→X[7]→X[8]
45: .0345→R[1,1];.0375→R[1,2];.0905→R[1,3];.09[illegible]→R[1,4];.095→R[1,5]
46: .102→R[1,6];.1065→R[1,7];.111→R[1,8];.1155→R[1,9];.12→R[1,10]
47: for J=11 to 16;R[1,J-1]+.005→R[1,J];next J
48: for J=1 to 16;(.23+.025(J-1))/2→R[2,J]
49: .2+.02(J-1)→R[3,J];.3+.03(J-1)→R[4,J];.25+.25(J-1)→R[5,J]
50: if R[5,J]>1;.767*R[5,J]+.233→R[5,J]
51: 1+(J-1)→R[6,J];.767*R[6,J]+.233→R[6,J];next J
52: for J=1 to 16;3.55+1.93(J-1)→R[7,J];77.5+10(J-1)→R[8,J];next J
53: for I=1 to 8;for J=1 to 15;(R[I,J+1]+R[I,J])/2→R[I,J];next J;next I
54: "start":int(BT)→Z;0→K→I;fxd 0;ina D,S,J,V,P
55: "data":buf "D";tfr 9,"D",1234;K+1→K
56: dsp "DATA SET #",K,"OF",S;ina rds("D")=1284
57: dsp "PROC #",K,"FILE #",J,"MAX",C[1]+C[2]
58: cll 'P1'(A[5]);cll 'P1'(A[6]);cll 'P1'(A);cll 'P1'(D);10snf(B,Z)→B
59: band(A,3)→A;band(C,3)→C
60: "loop":for J=1 to 10;for L=1 to 4;cll 'P2'(D);if J>2;jmp 5
61: if J=2;jmp 5
62: if L>2;jmp 4
63: int(D/100)→X[2L-1];100frc(D/100)→X[2L];if X[1]#30;wait 5000;gto "data"
64: jmp 2
65: D+1[L,J-2]→1[L,J-2]
66: if L=1;4-A→B;jmp 4
67: if L=2;if B<2;6→B;jmp 3
68: if L=2;5→B;jmp 2
69: L+4→B
70: for I=1 to 15;cll 'R2'(D);D+R[B,I]→R[B,I];next I
```

```
71: 1+[1,1]→J;next L;next J;1+E1→I; jmp 4
72: "01":0→P2; jmp 2
73: "02":rand(char(48),255)→P3;band(P3,15)+10ch£(P3,4)→P2
74: band(char(48),255)→P4;band(P4,15)+10ch£(P4,4)+10ch£2→P1;ret
75: wait 50;wrt 704,"A56";wait 50;wrt 722,"E1F4F2F3A0";for L=1 to 5
76: for I=1 to 2;for J=1 to 5;if L[I,J]>39;jmp 5
77: int £3.0;wrt 709,"C",L[1,J];wait 5
78: trg 722;fmt 6;red 722,P;P+P[I,J]→P[I,J]
79: if I=1;if J=1;F+V[1,1]→V[1,1];jmp 2
80: if I=1;if J=5;F+V[2,1]→V[2,1]
81: next J;next I;1+P→P
82: cll 'AVGDEGS'(P[1,2),J[1,2),6);cll 'AVGDEGS'(F[1,3],J[1,3],6)
83: next L;cll 'CALCA'
84: wrt 704,"A56"
85: wait 50;wrt 722,"F5F1F2F3A0";for J=1 to 3;if L[3,J]>39;jmp 3
86: fmt 2F3.0;wrt 709,L[3,J]+5,L[3,J];wait 5;trg 722;fmt 1;red 722,F
87: P+P[3,J]→P[3,J];wrt 709,"C";if P<100;if P>110;stp
88: next J;if X<4;goto "data"
89: fmt 2x,c10;red 708,A$
90: "*****GET CALCULATIONS*****":
91: for I=1 to 2;for J=1 to 5;if L[I,J]>39;jmp 2
92: F[I,J]/4→P[I,J]
93: next J;next I;for J=1 to 3;if L[3,J]>39;jmp 2
94: P[3,J]*5/4→P[3,J]
95: next J;P[1,5]→J[1,5];P[2,5]→J[2,5]
96: "BODE PLOT":.003692→R1;(P[3,1]-P[1,1])/R1→J[1,1]→J[3,1]
97: .8036→R1;for I=2 to 3;if L[3,I]>39;jmp 2
98: (P[3,I]-P[1,1])/R1→J[1,1]→J[3,I]
99: next I
100: for I=3 to 4;10(P[2,I]-1)→J[2,I];next I
101: 8P[1,4]→J[1,4]
102: 10P[1,1]→J[1,1]
103: 100P[2,1]→J[2,1]
104: for I=1 to 2;((V[I,3]-V[I,2]↑2/N)/(N-1))↑.5→V[I,1];next I
105: 5.14V[1,1]→V[1,1]
106: J[1,2]→J;J[1,1]→J;J[1,4]→J
107: √(V↑2+3↑2-2V3COS(E))→R1;.31R1→J[3,4]→J[3,0];VSIN(R)/R1→R2
```

```
103: (-S+Vcos(R))/r1→r3;if r3>0;acs(r2)→U[3,5];jmp 3
109: if r2>0;acs(r3)→U[3,5];jmp 2
110: 180-acs(r2)→U[3,5]
111: (U[3,5]+U[1,5])→U[3,5];cll 'SCALCALC'
112: for I=1 to 4;.25U[1]→A[1];next I;30.4U[5]→B[5];30.4A[6]→B[6]
113: for I=7 to 8;15.45*2A[I]→A[I];next I
114: for I=1 to 8;for J=1 to 15;if S[I,J]=0;1→S[I,J]
115: next J;next I
116: for I=1 to 8;for J=1 to 15;S[I,J]/A[I]→P[I,J];next J;next I
117: for I=7 to 8;for J=1 to 15;S[I,J]/(P[I-6,J]*W[I])→P[I,J];next J;next I
118: for I=1 to 4;for J=1 to 8;P[I,J]/1000/R→P[I,J];next J;next I
119: for I=1 to 8;for J=1 to 15;P[I,J]/(R[I,J+1]-R[I,J])→P[I,J]→O[I,J]
120: next J;next I
121: "sat":ina X,Y;for I=1 to 8;if I=1 or I=6;0→r1→r2→r3→r4→r5→r6→r7→r8→r9
122: for J=X[I]+1 to 15;1→S;if J<15;jmp 5
123: if I>1 and I<6;gto "nEXt"
124: if I=1;-1.5→X;gto "1"
125: if I=8 and r9≠0 and r7≠0;log(r8/r9)→X;log(r7/r9)→P;1→r5;gto "3"
126: 2.2→X;gto "1"
127: if O[I,J]=0 and I<6;gto "nEXt"
128: 4*π/3*E[I,J]↑3*O[I,J]→O[I,J]
129: if r9≠0;gto "extrap"
130: if O[I,J]=0 and I>5;gto "extrap"
131: log(O[I,J])→P;log(E[I,J])→X
132: if I=1 and J<9;gto "3"
133: if I<6;gto "2"
134: gto "3"
135: "extrap":r9+1→r9;r7+O[I,J]→r7;r8+E[I,J]→r8;gto "nEXt"
136: "3":r0+XX→r0;r1+X→r1;r2+1→r2;r3+PX→r3;r4+P→r4;if r5=0;gto "2"
137: X→B[9];P→B[10];1.5→X
138: "1":(r3r2-r4r1)/(r0r2-r1r1)→A[2]
139: (r0r4-r3r1)/(r0r2-r1r1)→A[1];B[1]+A[2]*X→P
140: "2":for K=0 to C;C+1-K→R;S+P[R,R]→P[R,R]
141: P+A[R]→A[R];P*X→P;SX→S
142: if K≠C;S+P[R-1,R]→P[R-1,R]
143: S*X→S;next K
144: "nEXt":next J;next I;if r5≠0;sfg 12
```

```
145: for I=1 to C+1
146: for K=1 to int((C+1)/2)
147: if I+1 and I+K<C+2 and I-K>0;F[I,I]→F[I-K,I+K]→F[I+K,I-K]
148: if I+K<=C+1 and I-K+1>0;F[I,I+1]→F[I+K,I-K+1]
149: if I+K<=C and I-K>0;F[I,I+1]→F[I-K,I+K+1]
150: next K;next I;inv F→F;mat FA→S
151: "plt":D→B;D+1→D;rcf 0,D,C[*];if D-1>C[1];trk 1;D-C[1]-2→E
152: rcf C;if flg7;gto "SKIP PRINT"
153: "out":wrt "1.7",C[3],D-1;if flg3;jmp 3
154: 80→Y[1];val(A$[1,2])→Y[2];val(A$[3,4])→Y[3];val(A$[5,6])→Y[4]
155: val(A$[7,8])→Y[5];val(A$[9,10])→Y[6]
156: wrt "1.3",Y[2],Y[3],Y[1],100Y[4]+Y[5],Y[6]
157: wrt "1.4",F,H;wrt "1.5",H[1,1];wtb M,10,13;if flg0;ret
158: fmt 11x,c5,16x,c9,11x,c14,z;wrt "1","Ship","True Wind","Relative Wind"
159: wtb M,13;cll 'SPACE'(7)
160: for I=1 to 3;for J=1 to 13;wtb M,95;next J;cll 'SPACE'(10);next I
161: fmt /,4x,c11,f5.1,7x,c11,f5.1,7x,c11,f5.1,;U[3,4]/.514→r1
162: wrt "1","Speed(kts):",U[1,4],"Speed(kts):",r1,"Speed(kts):",U[1,1]
163: fmt 7x,c8,f5.0,10x,c8,f5.0,10x,c8,f5.0,/
164: wrt "1","Heading:",U[1,3],"Heading:",U[3,5],"Heading:",U[1,2]
165: fmt 3x,c2,6x,c2,6x,c2,5x,c1,4x,c4,7x,c3,7x,c2,7x,c,z
166: wrt "1","Ts","Ta","Td","Tir1","Tir2","Z/L","RH","B"
167: fmt 6x,c,6x,c4,5x,c2,5x,c3,z;wrt "1","V","Vrms","U*","U*e"
168: wtb 1,13;for I=1 to 100;wtb 4,95;next I;wtb M,10,13
169: fmt x,f5.2,2x,f6.2,2x,f6.2,2x,f6.2,2x,f6.2,x,f10.2,2x,f6.2,4x,f5.1,z
170: Z[11]→r1;if flg11;U[2,2]→r1
171: wrt "1",U[3,3],U[3,2],U[3,1],U[2,3],U[2,4],Z[10],r1,B
172: fmt 2x,f5.1,3x,f5.1,2x,f6.3,2x,f6.3
173: wrt "1",U[1,5],U[2,5],Z[4],Z[8]
174: wtb 1,10,"Raw Counts",10,13;for J=1 to 15;for I=1 to 8
175: wrt "1.6",S[I,J];next I;wtb 1,10,13;next J;wtb M,10,13
176: wtb M,"dN/dr",10,13;for J=1 to 15;for I=1 to 8;wrt "1.6",F[I,J]
177: next I;wtb M,10,13;next J;wtb M,10,13
178: if not flg9;jmp 3
179: wtb M,"Radii";wtb 1,10,13;for J=1 to 15;for I=1 to 8;wrt "1.6",D[I,J]
180: next I;wtb 1,10,13;next J;wtb 1,10,13;clg 9
181: if not flg2;wtb 1,12;gto "SKIP PRINT"
```

```
182: sfg 0;wtb 3,12,13;cll 'out';cfg 0;wtb 1,27,65,-4,0,7,32,"log(dV/dr)"
183: wtb 1,27,65,1,56,7,32;wrt "1.3",C;5.01-1/6→Y
184: for I=1 to C+1;wtb 3,27,65,1,56,int(3Y/2),int(96Y)
185: wrt "1.9",B[I];Y-1/6→Y;next I
186: wtb 3,27,65,-4,-210,-1,-16,"log(radius)"
187: -1→X;-4→Y
188: wtb 1,27,65,int(15X/4),int(240X),0,0
189: if X≠0 and Xmod1=0;wtb 4,"!",10,8,8,8;wrt "%.1",X;gto +2
190: wtb 3,"-"
191: if (X+.05→X)<2.3;gto -3
192: wtb 1,27,65,0,0,int(1.5Y),int(96Y)
193: if Y≠0 and Ymod1=0;wrt "1.2","-",Y;gto +2
194: wtb 4,"!"
195: if (Y+.1→Y)<5.1;gto -3
196: for I=1 to 8;for J=X[I]+1 to 15
197: if R[I,J]=0 or D[I,J]=0;gto "NEXT"
198: log(R[I,J])→X;log(D[I,J])→Y
199: wtb 4,27,65,int(15X/4),int(240X),int(3Y/2),int(96Y)
200: wtb 1,B[I]
201: "NEXT":next J;next I;if not flg12;jmp 2
202: wtb 3,27,65,int(15B[9]/4),int(240B[9]),int(3B[10]/2),int(96B[10]),"e"
203: -1→X;0→Y;cfg 12
204: "crv":0→Y;if X>2;gto "belch"
205: for I=1 to C+1;YX+C[I]→Y;next I
206: Y→I[4];jmp 3
207: "belch":if X>2.1;if I[2]>0;I[6]→Y;jmp 2
208: I[4]+I[2]*(X-2)→Y→I[6]
209: if Y<-4 or Y>5;jmp 4
210: Y+6/96→Y
211: wtb 3,27,65,int(15X/4),int(240X),int(3Y/2),int(96Y)
212: wtb 4,"."
213: if (X+1/120→X)<2.2;gto "crv"
214: wtb 0,10,12,13
215: "SKIP PRINT":for I=1 to 4;for J=1 to 8
216: 32(I-1)→K;fts (H[I,J])→I$[K+4J-3,K+4J];next J;next I;for I=1 to 8
217: for J=1 to 15;60(I-1)→K;fts (P[I,J])→P$[K+4J-3,K+4J];next J;next I
218: for I=1 to 5;fts (Y[I])→Y$[4I-3,4I];next I
```

```
219: for I=1 to 3;for J=1 to 5;20(I-1)+K;fts (P[I,J])+T$[4J-3+K,4J+K]
220: fts (U[I,J])+U$[4J-3+K,4J+K];next J;next I
221: if flg7;lsp "FILE",D,"PRINT SUPPRESSED";jmp 2
222: dsp "RECORDING FILE",D-1,"MAX. NO.",C[1]+C[2]
223: if D>C[1]+C[2];cll 'tapemrk'
224: ref C,JS,K$,P$,J$,Y$,T,H,L,%[*];trk 0;fdf 0
225: "FLAGS":if flg1;cfg 1;gto "RADAR"
226: if flg5;cfg 5;lsp "Polv=",C," New=?";ent "",C;C+1+P;rdm A[R],G[R],F[R,R]
227: if flg6;cfg 6;lsp "Averaging time=",T,".New=?";ent "",T;int(3T)+%
228: if flg3;cfg 3;cll 'SENSOR STATUS'
229: gto "start"
230: "tapemrk":for I=1 to 5;beep;wait 500;next I
231: dsp "Insert new cassette,continue.";stp
232: dsp "Are you sure?Continue.";stp
233: ent "Tape number?",C[3]
234: trk 0;rew;mrk 129,950;rew;129+C[1]+C[2];1+D
235: ref 0,D,C[*];rew;trk 1;mrk 129,950;rew;trk 0;ret
236: "AVSOLVE":72a1+a1;a1-a2+a4;abs(a4)+a5
237: if a5>=180;(a4/a5)(a5-360)+a6;(a6/a3+a2)mod360+a2;jmp 2
238: a2+a4/a3+a2
239: 0+a1;ret
240: "SENSOR STATUS":prt "SENSOR STATUS";prt "DEACTIVATED"
241: prt Z$[1,2]&"/"&Z$[3,4]&" at "&Z$[5,8]
242: for I=1 to 3;for J=1 to 5;if L[I,J]>40;prt A$[5(I-1)+J]
243: next J;next I;ret
244: "SENSORS":ent "ACTIVATE(a),DEACTIVATE(d) OR (e)",C$;if C$="E";ret
245: if C$#"a" and C$#"d";beep;jmp -1
246: ent "SENSOR NAME",U$;if U$="E";jmp -2
247: for I=1 to 3;for J=1 to 5;if U$=A$[5(I-1)+J];jmp 2
248: next J;next I;beep;jmp -2
249: if C$="d";41+K[I,J];jmp -3
250: K[I,J]+L[I,J];jmp -4
251: "SIGMA":for I=1 to 2;V[I,1]/(L-1)+Q;Q+V[I,2]+V[I,2]
252: Q*Q+V[I,3]+V[I,3];0+V[I,1];next I;ret
253: "SCALCALC":Z[3]+r1;U[3,3]+r2;U[3,2]+r3;U[3,1]+r4;rad
254: if r1<.01;ina 3;20+Z[1];ret
255: if r1<2.2;.001*1.03*r1^(-.15)+r5;jmp 4
```

```
256: if r1<5;.001*(.77+.06*r1)→r5;jmp 3
257: if r1<8;.001*(.37+.057*r1)→r5;jmp 2
258: .001*(1.2+.025*r1)→r5
259: 10*exp(-.35/r5*.5)→r6;ln(z[11/r6)→r7;ln(z[11/.00002)→r8
260: cll 'p'(r2,z[2]);cll 'p'(r3,r20);if fl=11;.01*p(2,21*r20→z[3];jmp 2
261: cll 'c'(r4,z[3]);z[3]/r20*100→z[11]
262: .8*.35*z[11/(r3+273.15)*(.35*1.35/r8)/(.35/r7)↑2→r9
263: r9*(r3-r2+.01*z[11+.00061*(r3+273.15)*(z[3]-z[2]))/r1↑2→r9→r10
264: if r9>2.2;50→r13;cll 'pSI1'(r13,r11);cll 'pSI2'(r13,r12);gto 267
265: cll 'pSI1'(r10,r11);cll 'pSI2'(r10,r12);r9*(1-r11/r7)↑2/(1-r12/r8)→r13
266: fxd 3;disp r10,r13;if abs(r10-r13)>.001*abs(r9);r13→r10;jmp -1
267: r13→z[10];.35/r7*r1/(1-r11/r7)→z[4]
268: .35*1.35/r8*(r3-r2+.01*z[11)/(1-r12/r8)→z[5]
269: .35*1.35/r8*(z[3]-z[2])/(1-r12/r8)→z[6]
270: 40[1,5]√(.5140[1,1)↑2,1]√[1,1]→3
271: 4.4636-3*(.5140[1,1)↑.5(0[1,5]0[2,5])/3)↑3→z[7]
272: cll 'pHEPS'(r13,r14);(z[7]*.35*z[11/r14)↑.333→z[8]
273: ret
274: "pSI1":if r1>0;-4.7*p1→p2;ret
275: (1-15*p1)↑.25→p2;2*ln((1+p2)/2)+ln((1+p2↑2)/2)-2*atn(p2)+1.64→p2;ret
276: "pSI2":if r1>0;-6.5*p1→p2;ret
277: (1-3*p1)↑.5→p2;2*ln((1+p2)/2)→p2;ret
278: "pHEPS":if p1>0;(1+2.5*p1↑.67)↑1.5→p2;ret
279: (1+.5*abs(p1)↑.67)↑1.5→p2;ret
280: "c":.625*10↑(23.84-294.4/(p1+273.15)-5.03*log(p1+273.15))→p2;ret
281: "SPACE":wtb "1",32;jmp (p1-1→p1)=0
282: ret
*15632
```

c. AEROSOL 9835

```
1     ! P1G AEROSOL PROGRAM FULLDP 9835
10    OPTION BASE 1
20    COM Rv(0:0),Flag(0:15),C,K,J,SHORT G(10),B(9,52)
30    DIM O$[1284]
40      SHORT Q(1284),A(10),F(10,10),Ext(9),Gam(5)
50      SHORT Y(5),S(6,15),P(6,15),O(6,15),I(6),X(6),T(1),H(16)
60      SHORT R(6,16),N(6),D(6,15)
70      GOSUB Setup
80      DISP "INSERT DATA CARTRIDGE"
90      PAUSE
91      INPUT "NEW TAPE(NEW) OR OLD(O)",O$
92      IF O$="NEW" THEN GOSUB Markup
100      ASSIGN #9 TO "INDEX0"
110          READ #9;D,R,Tape_num
120          ASSIGN #9 TO *
130          Parts=2
140          PRINTER IS 16
150     INPUT "AVERAGING TIME IN MINUTES",T
160        Z=INT(1.5*T)
170     N(6)=N(4)=N(2)=N(1)=0
180        N(3)=0
190        N(5)=0
200     R(1,1)=.0845
210        R(1,2)=.0875
220        R(1,3)=.0905
230        R(1,4)=.094
240        R(1,5)=.098
250     R(1,6)=.102
260        R(1,7)=.1055
270        R(1,8)=.111
280        R(1,9)=.1155
290        R(1,10)=.12
300     FOR J=11 TO 16
310        R(1,J)=R(1,J-1)+.005
320        NEXT J
330     FOR J=1 TO 16
340        R(2,J)=(.23+.025*(J-1))/2
```

```
350     R(3,J)=.2+.02*(J-1)
360       R(4,J)=.3+.08*(J-1)
370       R(5,J)=.25+.25*(J-1)
380     IF R(5,J)>1 THEN R(5,J)=.767*R(5,J)+.233
390     R(6,J)=1+(J-1)
400       R(6,J)=.767*R(6,J)+.233
410       NEXT J
420     FOR I=1 TO 6
430       FOR J=1 TO 16-W(I)
440       R(I,J)=R(I,J+W(I))
450       NEXT J
460       NEXT I
470       FOR I=1 TO 6
480       FOR J=1 TO 15-W(I)
490       E(I,J)=(R(I,J+1)+R(I,J))/2
500       NEXT J
510       NEXT I
520     Start: Z=1.5*T
530       H=K=0
540       FIXED 0
550       MAT H=(0)
560       MAT B=(0)
570       MAT A=(0)
580       MAT Flag=(0)
590       DEFAULT OFF
600     Data:DISP "WAITING FOR DATA X-FER #";K;"FILE";D+1
610     ENTER Part$ FHS 1234 USING "#,1234A";25
620       K=K+1
630       Flag(2)=0
640       L=1
650       GOSUB R1
660       Y(5)=Pv1
670       L=L+1
680       GOSUB R1
690       A=Pv1
700       GOSUB R1
710       B=Pv1
```

```
720         B=10*SHIFT(B,4)
730      A=BINAND(A,3)
740         B=BINAND(B,3)
750         DISP Y(5),A,B
760 Loop: Flag(1)=0
770      FOR J=1 TO 20
780         Flag(1)=Flag(1) EXOR 1
790      FOR I=0 TO 15
800          GOSUB R2
810          Q=Pvl
820         IF I<>0 THEN GOTO Bin
830      IF NOT (J<3) THEN 870
840         Y(2*J-1)=INT(Q/100)
850         Y(2*J)=100*FRACT(Q/100)
860         IF Y(1)<>30 THEN GOTO Data
870      IF J<5 THEN GOTO Next_
880      H(J-4)=Q+H(J-4)
890         GOTO Next_
900 Bin: IF NOT Flag(1) THEN 930
910         S=A+1
920         GOTO 970
930      IF NOT (B<2) THEN 960
940         S=5
950         GOTO 970
960      S=6
970      S(S,I)=Q+S(S,I)
980 Next_: NEXT I
990         N(S)=1+N(S)
1000        NEXT J
1010        Flag(2)=Flag(2) EXOR 1
1020        IF NOT Flag(2) THEN 1040
1030        GOTO Loop
1040        IMAGE 11A,8A
1050        PRINT USING 1040;VAL$(Y(2))&"/"&VAL$(Y(3))&"/"&VAL$(Y(1)),VAL$(Y(4))&"
:"&VAL$(Y(5))
1060        H=H+1
1070     IF K<2 THEN GOTO Data
```

```
1080       DISP "file #";D+1;"Max file #";R
1090    GOSUB Fit
1100       GOTO Sp
1110 R1:Pv2=0
1120       GOTO 1160
1130 R2:  Pv3=BINAND(BINCMP(ROT(Q$[L,L])),255)
1140      L=L+1
1150       Pv2=BINAND(Pv3,15)+10*SHIFT(Pv3,4)
1160     Pv4=BINAND(BINCMP(ROT(Q$[L,L])),255)
1170       L=L+1
1180       Pv1=BINAND(Pv4,15)+10*SHIFT(Pv4,4)+100*Pv2
1190       RETURN
1200 Sp:E=D=D+1
1210        ASSIGN #9 TO "INDEX0"
1220         PRINT #9;D,R,Tape_num
1230         ASSIGN #9 TO *
1240    GOTO 1600
1250 Fit:FOR J=1 TO 15
1260       FOR I=1 TO 6
1270       IF NOT (I<>5) THEN 1290
1280       IF S(I,J)=0 THEN S(I,J)=1
1290    NEXT I
1300       NEXT J
1310        MAT P=S
1320    FOR I=1 TO 4
1330       N(I)=.25*N(I)
1340       NEXT I
1350       N(5)=30.4*N(5)
1360       N(6)=30.4*N(6)
1370    FOR J=1 TO 15
1380    N(J)=N(J)/(2000*N)
1390       T(1)=S(4,J)/N(4)
1400       S(4,J)=S(1,J)/N(1)
1410       S(1,J)=T(1)
1420    T(1)=S(3,J)/N(3)
1430       S(3,J)=S(2,J)/N(2)
1440       S(2,J)=T(1)
```

```
1450    T(1)=S(5,J)/N(5)
1460       S(5,J)=S(6,J)/N(6)
1470       S(6,J)=T(1)
1480       NEXT J
1490       N(15)=N(15)/(2000*N)
1500    FOR I=1 TO 5
1510       X(I)=N(I)
1520       NEXT I
1530    FOR I=1 TO 5
1540       FOR J=1 TO 15-A(I)
1550       S(I,J)=S(I,J+A(I))/(R(I,J+1)-R(I,J))
1560    NEXT J
1570       NEXT I
1580    MAT O=S
1590    RETURN
1600    GOSUB Poly
1610     ASSIGN #9 TO "ASCII"
1620          PRINT #9,S;S(*),N(*),Y(*),X(*),P,Ext(*),Gam(*),S(*)
1630          ASSIGN #9 TO *
1640          PRINTER IS 16
1641          IF D=R THEN GOSUB Markup
1650       GOTO Start
1660 Prt:IMAGE #,6D.2D,1X,4D.2DE
1670       PRINT USING 1660;(R(I,J+1)+R(I,J))/2,S(I,J)
1680       RETURN
1690 Setup:DEG
1700    K=61 setup
1710       A =1
1720        ASSIGN #9 TO "HISCOP"
1730         READ #9;S(*)
1740    IMAGE 3D
1750    IMAGE #,K,3D
1760    IMAGE "Date",3D,"/",2D,"/",2D,"  Time",3D,":",2D," at end of run"
1770    IMAGE /,"Averaging time = ",3D," minutes"
1780    IMAGE "Probe voltage  A = ",2D.3D," volts"
1790    IMAGE #,1X,D.2DE
1800    IMAGE "Tape #",3D," File #",3D
```

```
1810    IMAGE #,"Polynomial of order ",2D
1820    IMAGE #,1X,D.7DE
1830    J=7
1840    I=J+1
1850       READ A(I),J(I),P(I,I)
1860    DISP "printer on,set form, push cont"
1861    PAUSE
1870    WRITE BIN (1;27,69,27,84,32,32,32,32,27,77,27,76,15,0,14
1880    WRITE BIN (1;27,79,15,48,6,32
1890    RETURN
1900 Poly:  FOR I=1 TO 3
1910    FOR J=1 TO 3
1920     O(I,J)=0
1930     NEXT J
1940     NEXT I
1950     O(5,1)=O(6,1)=0
1960    FOR J=1 TO 15
1970     O(4,J)=0
1980     NEXT J
1990     Jcont=Jacc=Gcont=Vacc=Vracc=Vljracc=0
2000     R0=R1=R2=R3=R4=0
2010      IAF A=ZER
2020       IAF P=ZER
2030   FOR I=1 TO 5
2040     FOR J=1 TO 16
2050     J=1
2060     IF (I=1) AND (J=16) THEN GOTO Left
2070     IF (I=6) AND (J=16) THEN GOTO Right
2080     IF J>15 THEN GOTO Nextt
2090     IF O(I,J)=0 THEN GOTO Nextt
2100     P=G*LGT(O(I,J))
2110     X=LGT(E(I,J))
2120     IF I<3 THEN GOSUB Junje
2130     IF I>2 THEN GOSUB Gamma_it
2140     IF (I=1) AND (J<9) THEN GOTO Three
2150     IF I=6 THEN GOTO Three
2160    GOTO Two
```

```
2170 Left:   X=-1.5
2180   GOTO One
2190 Right:X=1.5
2200   GOTO One
2210 Three:R0=R0+X*X
2220     R1=R1+X
2230     R2=R2+1
2240     R3=R3+P*X
2250     R4=R4+P
2260   GOTO Two
2270 One:P=((R3*R2-R4*R1)*X+R0*S4-R3*R1)/(R0*R2-R1*R1)
2280     R0=R1=R2=R3=R4=0
2290   PRINT USING "DD.DD,XX,DD.DD";X,P
2300     IF I=6 THEN G=5
2310 Two:FOR K=0 TO C
2320     P(C+1-K,C+1-K)=G+P(C+1-K,C+1-K)
2330     A(C+1-K)=P+A(C+1-K)
2340     P=P*X
2350     G=G*X
2360     IF K<>C THEN P(C-K,C+1-K)=G+P(C-K,C+1-K)
2370     G=G*X
2380     NEXT K
2390 Nextt:NEXT J
2400     NEXT I
2410   FOR I=1 TO C+1 ! poly
2420   FOR K=1 TO INT((C+1)/2)
2430   IF NOT ((I<>1) AND (I+K<C+2) AND (I-K>0)) THEN 2450
2440      P(I+K,I-K)=P(I-K,I+K)=P(I,I)
2450   IF (I+K<=C+1) AND (I-K+1>0) THEN P(I+K,I-K+1)=P(I,I+1)
2460   IF (I+K<=C) AND (I-K>0) THEN P(I-K,I+K+1)=P(I,I+1)
2470   NEXT K
2480      NEXT I
2490   MAT P=INV(P)
2500      MAT G=P*A
2510      GOSUB Gamma_cal
2520   IF Flag(7) THEN GOTO DEF
2530   OUTPUT 1_ USING 1800;Tape_num,D! out
```

```
2540    OUTPUT 1 USING 1760;X(2),X(3),X(1),X(4),X(5)
2550    OUTPUT 1 USING 1770;P
2560    OUTPUT 1 USING 1780;H(1)
2570    WRITE BIN(1);10,13
2580    OUTPUT 1 USING "#,K,3,3";"Raw counts",10,13
2590       FOR J=1 TO 15
2600       FOR I=1 TO 6
2610    OUTPUT 1 USING 1790;F(I,J)
2620          NEXT I
2630          WRITE BIN(1);10,13
2640          NEXT J
2650          WRITE BIN(1);10,13
2660    OUTPUT 1 USING "#,K";"dN/dr"
2670    WRITE BIN(1);10,13
2680    FOR J=1 TO 15
2690    FOR I=1 TO 6
2700    IF DER (J>15-X(I)) THEN 2730
2710       OUTPUT 1 USING "#,10X"
2720       GOTO 2740
2730    OUTPUT 1 USING 1790;S(I,J)
2740    NEXT I
2750    WRITE BIN(1);10,13
2760    NEXT J
2770    WRITE BIN(1);10,13
2780  OUTPUT 1 USING "/,15A,MDDD.DD";"JUNGI COEF=",Junge
2790  OUTPUT 1 USING "15A,MDDD.DD";"GAMMA ZERO=",Gamma0
2800  OUTPUT 1 USING "15A,MDDD.DD";"BETA =",Beta
2810  OUTPUT 1 USING "15A,ZD.DDE";"N ZERO=",No
2820    OUTPUT 1 USING "#,3,#,3,3,B,3,K";27,65,INT(-3.75),INT(-240),0,16,"log(rad
ius)"
2830    OUTPUT 1 USING "#,#,3,3,3,3,3,K";27,65,INT(-1.35),INT(-120),7,32,"log(dN/
dr)"
2840    WRITE BIN(1);27,65,1,56,7,32
2850    OUTPUT 1 USING 1910;C
2860       Y=5.0E-1/6
2870    FOR I=1 TO C+1
2880    WRITE BIN(1);27,65,1,50,INT(3*Y/2),INT(96*Y)
```

```
2890    OUTPUT 4 USING 1320;S(I)
2900       Y=Y-176
2910    NEXT I
2920 Plt: !
2930    X=-1
2940       Y=-4
2950    WRITE BIN(4;27,65,INT(15*X/4),INT(240*X),0,0
2960    IF NOT ((X<>0) AND (X MOD 1=0)) THEN 3000
2970       OUTPUT 4 USING "#,K,2,3,3,3";"!",10,3,8,3
2980       OUTPUT 4 USING 1740;X
2990       GOTO 3010
3000    OUTPUT 4 USING "#,K";"-"
3010    X=X+.05
3020       IF X<2.05 THEN GOTO 2950
3030    WRITE BIN(4;27,65,0,0,INT(1.5*Y),INT(96*Y)
3040    IF NOT ((Y<>0) AND (Y MOD 1=0)) THEN 3070
3050       OUTPUT 4 USING 1750;"-",Y
3060       GOTO 3030
3070    OUTPUT 4 USING "#,K";"!"
3080    Y=Y+.1
3090       IF Y<5.1 THEN GOTO 3030
3100    Flag(5)=0
3110       FOR I=1 TO 5
3120    FOR J=1 TO 15-S(I)
3130    IF (E(I,J)=0) OR (O(I,J)=0) THEN GOTO Next
3140    X=LGT(E(I,J))
3150       Y=LGT(O(I,J))
3160    WRITE BIN(4;27,65,INT(15*X/4),INT(240*X),INT(3*Y/2),INT(96*Y)
3170    IF NOT Flag(5) THEN 3200
3180       OUTPUT 4 USING "#,K";"O"
3190       GOTO Next
3200    OUTPUT 4 USING "#,K";"*"
3210 Next:NEXT J
3220       Flag(5)=Flag(5) EXOR 1
3230       NEXT I
3240    X=-1
3250       Y=0
```

```
3250 Drv:Y=0
3270      FOR I=1 TO C+1
3280   Y=Y*X+C(I)
3290   NEXT I
3300   IF (X<-1) OR (X>5) THEN GOTO 3340
3310   Y=Y+5/95
3320   WRITE BIN 1;27,65,INT(15*X/4),INT(240*Y),INT(3*Y/2),INT(96*Y)
3330   OUTPUT 1 USING "#,B";","
3340   X=X+1/120
3350      IF X<1.1 THEN GOTO Drv
3360 Brack: !
3370      C=-1.2
3380      Z=-.3
3390   WRITE BIN 1;27,65,INT(15*Z/4),INT(240*Y),INT(3*Z/2),INT(96*Z)
3400   OUTPUT 1 USING "#,B";"Wavelength  Extinction"
3410 Eff:   FOR K=1 TO 9
3420      J=2
3430      Z=0
3440      Par=-1
3450      GOSUB Effof
3460      A=Z*2
3470   Flag(12)=0
3480      FOR I=-1 TO 1.4 STEP .1
3490         L=10^(I+.05)
3500         D=10*I
3510         G=L-D
3520   P=10^(I+.1)
3530         W=P-D
3540         M=G/W
3550         X=A*V*V/(6*(P-D))
3560   Y=V/2-X*X/3
3570         J=J+1
3580         Par=I+.05
3590         GOSUB Effof
3600         B=Z*2
3610         J=J+1
3620   Par=I+.1
```

```
3630    GOSUB Effof
3640    F=Pv2
3650       Z=Z+(V-X-Y)*A+B*X+Y*F
3660       A=F
3670       IF Flag(12) THEN GOTO 3690
3680    NEXT I
3690    Ext(K)=Z
3700    U=B(K,1)
3710    PRINT USING "DDD.DDD,3X,SD.DDE";U,Z
3720       S=S-1/6
3730       IF Flag(7) THEN GOTO 3770
3740    WRITE BIN 1;27,65,INT(15*S/4),INT(240*S),INT(3*S/2),INT(96*S)
3750    IMAGE #,5D.4D,SD.3D,
3760    OUTPUT 1 USING 3750;U,Z
3770    NEXT K
3780       IF Flag(7) THEN GOTO 3800
3790    WRITE BIN 1;12,13
3800    RETURN
3810 Effof:  Pv1=Par
3820     Pv2=0
3830    FOR L=1 TO C+1
3840     Pv2=Pv1*Pv2+C(L)
3850     NEXT L
3860    IF (Pv1>.5) AND (Pv2>0) THEN Flag(12)=1
3870    T=Pv2
3880     Pv2=B(K,J)*PI*10^(Pv2+2*Pv1-6)
3890    RETURN
3900 Junge:Jcont=Jcont+1
3910     Jacc=Jacc+4*PI/3*E(I,J)^4*O(I,J)
3920    RETURN
3930 Gamma_ft:Gcont=Gcont+1
3940     Vol=4*PI/3*E(I,J)^3*O(I,J)
3950     Delr=R(I,J+1)-R(I,J)
3960     Vacc=Vacc+Vol*Delr
3970     Vracc=Vracc+Vol*E(I,J)*Delr
3980     Vlgracc=Vlgracc+Vol*LOG(E(I,J))*Delr
3990    RETURN
```

```
4000 Gamma_cal:Gam(1)=Junge=Jacc/Jcont
4010    A=LOG(Vracc/Vacc)-Vlgracc/Vacc
4020    Gam(2)=Gammaz=(1+(1+4*A/3)^.5)/4/A
4030    Gam(3)=Beta=Vracc/Vacc/Gammaz
4040    B2=EXP(-Gammaz)*Gammaz^(Gammaz-.5)*(2*PI)^.5
4050    Gam(4)=Gofz=B2*(1+(12*Gammaz)^(-1)+(288*Gammaz^2)^(-1)-139/(51840*Gammaz
^3))
4051    Fgacc=Ffacc=0
4060   FOR I=3 TO 6
4070   FOR J=1 TO 15
4080    IF D(I,J)=0 THEN GOTO Summ
4090    Fg=E(I,J)^(Gammaz-1)*EXP(-E(I,J)/Beta)/(Gofz*Beta^Gammaz)
4100    Vol=4*PI/3*E(I,J)^3*D(I,J)
4110    Delr=R(I,J+1)-R(I,J)
4120    Fgacc=Fgacc+Vol*Delr*Fg
4130    Ffacc=Ffacc+Fg^2*Delr
4140 Summ:NEXT J
4150   NEXT I
4160    Gam(5)=No=Fgacc/Ffacc
4170   RETURN
4180 Tarkun:DISP "INSERT NEW TAPE"
4190   PAUSE
4200   INITIALIZE ":T15"
4210   REMID
4220   CREATE "INDEX0",1,24
4230   CREATE "AERO",360,530
4231   D=0
4232   R=360
4233   Tape_num=Tape_num+1
4234   ASSIGN #1 TO "INDEX0"
4235   PRINT #1;D,R,Tape_num
4236   ASSIGN #1 TO *
4240   REMID
4250   RETURN
```

2. Aircraft Programs

a. Meteorology 9835

```
1     ! STRIP-FF DATA ACQUISITION PROGRAM TST 9335
10    OPTION BASE 1
20    OVERLAP
30    DEFAULT ON
40    LOCAL LOCKOUT 7
50    File_number=1
60    DIM T1$[14],T2$[14],Chan$(20)[3],Relay$(3)[3],SHORT Data(70,20)
70    DIM A$[14]
80    FIXED 0
90    FOR I=0 TO 9
100   Chan$(I+1)="0"&VAL$(I)
110   Chan$(I+11)="1"&VAL$(I)
120   NEXT I
130   FOR R=0 TO 2
140   Relay$(1+R)=VAL$(30+R)
150   NEXT R
160   Scanner=709
170   Multimeter=722
180   IMAGE 10A
190   OUTPUT Multimeter USING 180;"F1R3I2T3A1"
200   ON KEY #0,2 GOTO Cycle
210   ON KEY #1,1 GOSUB Key_burst
220   ON KEY #2,1 GOSUB Key_clock
230   C=1
240 Cycle:  ON INT #7 GOSUB Tape
250   IAP Data=&SR
260   Mode=1
270   SENDBUS 7;"?)U";"3,"&Relay$(1)&"E"
280   OUTPUT 9;"R"
290   ENTER 9;T1$
300   ASSIGN #1 TO "DATA"&VAL$(File_number)
310   BUFFER #1
320   PRINT #1;T1$,END
330   FOR L=1 TO 70
340   IF L=4 THEN SENDBUS 7;"?)U";"3,"&Relay$(2+File_number MOD 2)&"E"
350   FOR Count=1 TO 2
360   FOR K=1 TO 19
```

```
370  SEND BUS 7;"?)0";Chans(K)&"D"
380  TRIGGER Multimeter
390  ENTER Multimeter USING"#,B";Sample
400  IF K<>2 THEN GOTO Next
410  FIXED 0
420  DISP "C=";C;
430  FIXED 4
440  DISP "V=";Data(L,C);
450  FIXED 0
460  DISP "    F=";File_number;"  S=";L
470  Next: NEXT K
480  NEXT Count
490  GOTO Next1
500  Take: Data(L,K)=Sample+Data(L,K)
510   IF Count=1 THEN GOTO 540
520  Data(L,K)=Data(L,K)/2
530  PRINT #1;Data(L,K),END
540  RETURN
550  Next1:  Data(L,20)=0
560  PRINT #1;Data(L,20),END
570  NEXT L
580  OUTPUT 9;"R"
590  ENTER 9;F2$
600  PRINT #1;F2$,END
610  ASSIGN #1 TO *
620  File_number=File_number+1
630  On Mode GOTO Check,Burst,Clock
640  Check:  IF File_number<33 THEN GOTO Cycle
650   GOTO Finish
660  Key_burst:  Mode=2
670   RETURN
680  Key_clock:  Mode=3
690   RETURN
700  Burst:  INPUT "INPUT BURST CHANNEL 2, 8",S
710   OFF INT #7
720   OUTPUT 9;"R"
730   ENTER 9;F1$
```

```
740   SEND@SUB 7;"?)J";Chan$(2)&"E"
750   FOR L=1 TO 70
760   FOR K=1 TO 19
770   TRIGGER Multimeter
780   ENTER Multimeter USING 15;Data(L,K)
790   IF (L MOD 10<>0) OR (K<>1) THEN GOTO Next_K
800   FIXED 0
810   DISP "BURST";3;
820   FIXED 4
830   DISP "VOLTS ";Data(L,K);
840   FIXED 0
850   DISP "SCAN";L
860 Next_K:  NEXT K
870   Data(L,20)=3
880   NEXT L
890   OUTPUT 9;"R"
900   ENTER 9;T2$
910   ASSIGN #1 TO "DATA"&VAL$(File_number)
920   PRINT #1;T1$,Data(*),T2$,END
930   ASSIGN #1 TO *
940   File_number=File_number+1
950   FOR I=1 TO 5
960   BEEP
970   WAIT 100
980   NEXT I
990   IF File_number<38 THEN GOTO Burst_continue
1000 GOTO Finish
1010 Burst_continue: INPUT "CONTINUE BURST MODE N=0 or Y=1",Reply
1020 IF Reply=1 THEN GOTO Burst
1030 IF Mode=3 THEN GOTO Clock
1040 GOTO Cycle
1050 Clock: OUTPUT 9;"R"
1060 ENTER 9;P$
1070 DISP P$
1080 GOTO Clock
1090 Finish: FOR I=1 TO 10
1100 BEEP
```

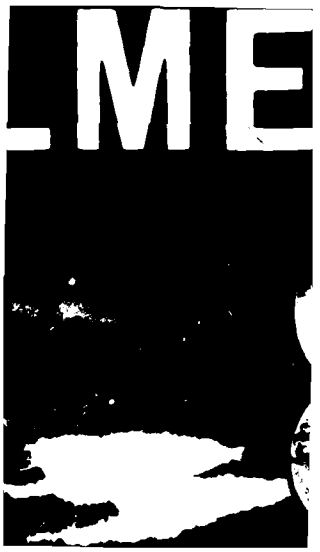
```
1110 READ 100
1120 [illegible]
1130 STOP "ERROR NEW DATA CARD/PAGE"
1140 END
```

b. Aerosol

```
32: out "N";ter 2,"N",30
33: if ins("N")=30
34: dep "checkin synchro"
37: 255-num(a$[4,4])*2;snd(2,1)*2
38: 255-num(a$[44,44])*2;snd(2,1)*2
39: if z=2;0*1;jmp 2
40: 1*1
41: jmp 2
42: frt 2,2,1
43: ret
44: lbf 0
*13200
```

INITIAL DISTRIBUTION LIST

		DODAAD Code	No. of Copies
1.	Defense Documentation Center Cameron Station, Building 5 Alexandria, Virginia 22314	S47031	12
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940		2
3.	Dean of Research, Code 012 Naval Postgraduate School Monterey, California 93940	N62271	1
4.	C. W. Fairall, The BDM Corporation Naval Postgraduate School Monterey, California 93940		5
5.	Associate Professor K. L. Davidson Code 63Ds Naval Postgraduate School Monterey, California 93940		1
6.	Associate Professor G. Schacher Code 61Sq Naval Postgraduate School Monterey, California 93940		1
7.	Director, Naval Research Laboratory Attn: Code 2627 Washington, D.C. 20375	N00173	6
8.	Office of Naval Research Branch Office 1030 East Green Street Pasadena, California 91106	N62879	1
9.	Mr. Robin Simpson Office of Naval Research Resident Representative Stanford University, Room 165 Durond Aeronautics Building Stanford, California 94305		1


LME